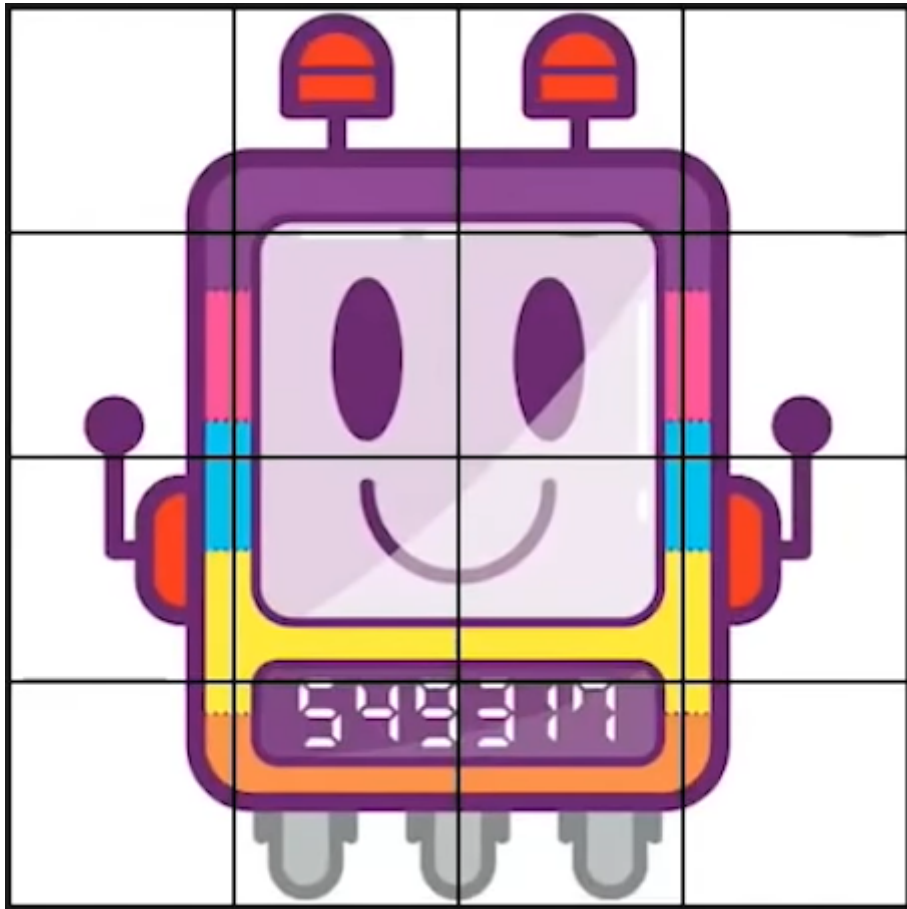


## Undervisning 22 - Slide puzzle



Jeres mål med denne opgave er at lave et klassisk slide-puzzle spil, hvor man har 16 dele af et større billede og skal dermed trykke på dem for at rykke brikkerne på plads og danne det fulde billede. (Se overstående billede)

## Del 0 - Setup

0.1

## Lav et nyt github repository

0.2

Lav nu i jeres repo, et nyt p5.js projekt i visual studio code

0.3

Skub jeres tomme projekt op til github.

0.4

Som minimum skal i skubbe hver del af opgaven som i laver op til github umiddelbart efter det er lavet. I må også gerne gøre det oftere.

## Del 1 - Billedet

*Skriv alle disse opgaver som pseudokode først og sammenlign med en sidemand hvad i har tænkt. Derefter implementer det i kode.*

### 1.1

Planlæg en struktur for jeres program og lav nu et klasse diagram for at illustrere det. (I må gerne delvist starte og komme tilbage her efterfølgende, hvis det er svært at gennemskue hele programmets opbygning fra starten.)

### 1.2

Lav en ny klasse *Tile* og indlæs de 16 billeder ind som *Tiles* og placer dem i et array

### 1.3

Lav en funktion der ud fra dit array af billeder, tegner dem alle på skærmen i rækkefølge, således der nu er tegner et helt billede på skærmen. Tilføj en smule mellemrum mellem hver *Tile*, således man kan se de er separat.

### 1.4

Fjern sidste brik af dine *Tiles* således kun 15 tegnes i alt.

### 1.5

Skub et nyt element til dit array der repræsenterer det tomme felt.

### 1.6

Bland nu dit array af *Tiles* således de bliver tegnet i tilfældig rækkefølge.

## Del 2 - Gameplay

### Beskrivelse af game loop.

Når spilleren klikker på skærmen, bliver der tjekket hvorvidt der er et felt ved siden af det som er klikket på, der er tomt. Hvis et nabo felt er tomt, så byt det kikkede felt med nabo feltet. Når alle felter er placeret korrekt stopper spillet.

### 2.1

Lav et flow diagram over spillets game-loop.

### 2.2

Implementer dit flow diagram i kode. Prøv at tage det et trin af gangen.

### 2.3

Hvis ikke du gjorde det som en del af 2.2, så implementer nu en funktion *isSolved* der tjekker om hvorvidt alle brikkerne er på sin rigtige plads.

*Hint: Dette kræver eventuelt en udvidelse til klassen *Tile*, så de indeholder information om hvad deres rigtige plads er.*

## Del 3 - Billede splitning

I stedet for at upload 16 pre-splittede dele af 1 billede, er det muligt at indlæse 1 enkelt billede og i koden dele det op i antal ønskede stykker.

### 3.1

Gør brug af funktionen *Image. copy* fra p5.js til manuelt at dele dit billede op.

<https://p5js.org/reference/#/p5/copy>.

### 3.2

Ændrer nu i dit program således du kan definere lige hvilken kvadratisk størrelse spilleflade du ønsker, og dit billede bliver dermed delt efter det samt spillet også virker.

### 3.3

Gå nu tilbage til jeres klassediagram og tilføj/ret eventuelle udvidelser lavet undervejs

## Del 4 - Ekstra features

### 4.1

Tilføj en *move-counter* i hjørnet af dit spil, der tæller hvor mange gange spilleren har flyttet en brik.

### 4.2

Tilføj nu en vinder skærm der viser antal brugte ture samt ønsker spilleren tillykke med sejren.

### 4.3

Gå nu tilbage til jeres klassediagram og tilføj/ret eventuelle udvidelser lavet undervejs